

AMENDMENTS TO THE CLAIMS:

This listing of claims will replace all prior versions and listings of claims in the application:

1. (Currently Amended) A computer-implemented method (400) for dynamic data type enrichment comprising the steps:

using (410) at least one basic data type (110) in a predefined application program (210); and

adding (420) metadata (150) to the at least one basic data type (110) at runtime when the application program (210) is executed to create a variable in the application program that comprises both the basic data type and the metadata.

2. (Currently Amended) The method (400) of claim 1, wherein the application program (210) uses an application programming interface (190) for accessing (415) the metadata (150) before adding the metadata to the basic data type (420).

3. (Currently Amended) The method (400) of claim 2, wherein the application program (210) calls through the application programming interface (190) at least one metadata service (191) that relates to the basic data type (110).

4. (Currently Amended) The method (400) of claim 3, wherein the at least one metadata service (191) copies the metadata (150) to a metadata cache.

5. (Currently Amended) The method (400) of claim 1, ~~anyone of the claims 1 to 4~~, wherein the basic data type (110) is defined in a programming language used by the application program (210).

6. (Currently Amended) The method (400) of claim 5, wherein the metadata (150) is associated with a specific data type (120) defined in a metadata store (210).

7. (Currently Amended) The method (400) of claim 6, wherein the application program (210) provides a mapping (302) between the specific data type (120) and the basic data type (110).

8. (Currently Amended) The method (400) of claim 6, wherein the application program uses a variable (201) to map (302) the specific data type (120) to the basic data type (110).

9. (Currently Amended) The method (400) of claim 6 ~~any of the claims 6 to 8~~, wherein the metadata (150) is stored in a private instance of the metadata store together with the application (220).

10. (Currently Amended) The method (400) of claim 6 ~~any of the claims 6 to 8~~, wherein the metadata (150) is stored in a shared instance of the metadata store (220).

11. (Currently Amended) A computer program product comprising instructions ~~that when loaded into~~ embodied on a memory of a computer system (900) ~~that~~ cause at least one processor of the computer system (900) to execute the steps of claim 1 ~~anyone of the claims 1 to 10~~.

12. (Currently Amended) A computer system (900) comprising:
a memory storing an application program (210) that uses a basic data type (110);
and
a processor executing instructions to add metadata (150) to the basic data type (110) when executing the application program to create a variable in the application program that comprises both the basic data type and the metadata (210).

13. (Currently Amended) The computer system (900) of claim 12 further comprising an application programming interface (190) to access (415) the metadata (150) from the application program (210).

14. (Currently Amended) The computer system (900) of claim 13, wherein the application programming interface (190) provides at least one metadata service (191) that relates to the basic data type (110) used by the application program (210).

15. (Currently Amended) The computer system (900) of ~~anyone of the claims 12 to 14~~ claim 14, further comprising a metadata cache, the at least one metadata service (191) copying the metadata (150) to the metadata cache.

16. (Currently Amended) The computer system (900) of claim 12 ~~anyone of the claims 12 to 15~~, wherein the basic data type (110) is defined in a programming language used by the application program (210).

17. (Currently Amended) The computer system (900) of claim 16, wherein the metadata (150) are associated with a specific data type (120) defined in a metadata store (210).

18. (Currently Amended) The computer system (900) of claim 17, wherein the application program (210) provides a mapping (302) between the specific data type (120) and the basic data type (110).

19. (Currently Amended) The computer system (900) of claim 18, wherein the application program uses a variable (201) to map (302) the specific data type (420) to the basic data type (410).

20. (Currently Amended) The computer system (900) of claim 17 ~~anyone of the claims 17 to 19~~, wherein the metadata (150) is stored in a private instance of the metadata store together with the application (220).

21. (Currently Amended) The computer system (900) of claim 17 ~~anyone of the claims 17 to 19~~, wherein the metadata (150) is stored in a shared instance of the metadata store (220).

22. (Withdrawn) A method for generating an application program (210) comprising the steps:

making available at least one metadata service (191) to be used in the application program (210) at design time for defining how the application program (210) can access metadata (150) at runtime; and

including a first implementation portion of the least one metadata service (191) in the IDE (800) that is unaffected by changes of a second implementation portion of the least one metadata service (191) in a metadata store (220).

23. (Withdrawn) An integrated development environment (IDE) (800) for generating an application program (210) by performing the steps of claim 22.

24. (Currently Amended) A method for changing metadata (150) comprising the steps:

executing an application program (210) that uses at least one metadata service (191) to access the metadata (150) in a metadata store (220);

changing the metadata (150) in the metadata store (220) at runtime of the application program (210); and

using the at least one metadata service (191) in the application program (210) for using the changed metadata without restarting the application program to create a variable in the application program that comprises both a data type and the metadata (210).